Peach: A Multicore Communication System on Chip with PCI Express

著者	Otani Sugako, Kondo Hiroyuki, Nonomura Itaru, Hanawa Toshihiro, Miura Shin'ichi, Boku Taisuke
journal or publication title	IEEE micro
volume	31
number	6
page range	39-50
year	2011-11
権利	(C) 2011 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.
URL	http://hdl.handle.net/2241/115373

doi: 10.1109/MM.2011.93

PEACH: A MULTICORE COMMUNICATION SOC WITH PCIe

Sugako Otani[1], Hiroyuki Kondo[1], Itaru Nonomura[1]
Toshihiro Hanawa[2], Shin'ichi Miura[2], Taisuke Boku[2]
[1]Renesas Electronics Corporation
[2]University of Tsukuba

THE EIGHT-CORE COMMUNICATION SOC, CODE-NAMED “PEACH”, WITH FOUR 4X PCI EXPRESS REV.2.0 PORTS, REALIZES A HIGH PERFORMANCE, POWER-AWARE, HIGHLY DEPENDABLE NETWORK. THE NETWORK USES PCI EXPRESS NOT ONLY FOR CONNECTING PERIPHERAL DEVICES BUT ALSO AS A COMMUNICATION LINK BETWEEN COMPUTING NODES. THIS APPROACH OPENS UP NEW POSSIBILITIES FOR A WIDE RANGE OF COMMUNICATIONS.

Keywords: PCI Express, Network communications, Multicore/single-chip multiprocessors, Energy-aware systems, Dependability

Recent trends using computing clusters point to a growing demand for high compute density environments in various application fields such as server appliances including distributed web servers. Distributed web servers need many server nodes and low-latency/high-bandwidth network for operating a massive amount of web services including distribution of high-definition movies. In these computing clusters, power consumption and system cost have increased. Therefore, it is vital to downsize computing cluster without loss of high dependability including fault tolerance.

To realize high performance, power-aware and highly dependable network, we have proposed a small computing cluster for embedded systems. This small computing cluster has low power computing nodes such as PCs and embedded CPUs. It also has a communication link between the computing nodes called PEARL (PCI Express Adaptive and Reliable Link) [1].

Commodity network devices such as Gigabit Ethernet (GbE) and InfiniBand are not sufficient for small computing clusters. InfiniBand is a switched fabric communication link used in high-performance computing and enterprise data centers. It achieves high reliability but power consumption is relatively high [2]. GbE is a cost and power rival of InfiniBand. However, GbE does not match transmission performance of InfiniBand.

In order to achieve both high performance and low power consumption, PEARL uses PCI Express

(PCIe) [3], which is a high-speed serial I/O interface standard in PCs, not only for connecting peripheral devices but also as a communication link between computing nodes.

To implement PEARL, a communication device, PEACH (PCI Express Adaptive Communication Hub) which acts as a switching device, has been developed. PCIe transfers packets point-to-point bidirectionally with high bandwidth. However it connects only between a Root Complex (RC) and Endpoints (EPs).

Therefore, a problem exists in that PCIe interfaces on PCs cannot be connected to each other, because every node CPU in the computing node is an RC. To solve this problem, each node CPU is equipped with a network interface card with PEACH. The node CPUs are connected to each other by a PCIe cable. To pair an RC with EPs at each end of the PCIe cable, PEACH can switch the RC port and EP ports to connect two computing nodes peer-to-peer.

PEACH can address two computing nodes as peers, breaking the traditional PCIe limit of only linking to a single master.

PEACH overview

The communication device, PEACH, has four PCIe Rev.2.0 ports with four lanes each, and employs an eight-core control processor [4]. There are several advantages of using of PEACH in the proposed network. Four PCIe ports can broaden the scope of selection of network topology. The high bandwidth of 20Gbps/port is equal to that of InfiniBand DDR 4x. The multicore control processor performs fault handling and system monitoring / logging for dependability. The multicore processor also controls the network system for power awareness. Figure 1 shows a prototype of a PEARL network system. PEACH behaves as a communication interface to other computing nodes as well as a communication switch.

How PEACH works in a dependable network

In Figure 2a) PEACH connects four nodes of the network via its four PCIe ports. One of the adjacent nodes is a node CPU and the others are PEACHs. When a request is received from a node CPU via the PCIe port, PEACH #0 generates a packet header, which is then the packet sent to the appropriate destination port. When a packet is received from a node, PEACH #1 analyzes the packet header, and PEACH forwards the packet to another node or passes it to the node CPU.

The functions for error detection, flow control, and retransmission control in the PCIe specifications

are processed by the hardware automatically. InfiniBand only supports link-by-link cut off, so only one faulty lane must causes a link to go down. In contrast, when a link error that cannot be automatically corrected occurs, PEACH reduces the number of lanes of the PCIe port to remove the defective lane by re-initializing the link. This enhancement from InfiniBand can provide higher reliability on the network.

The detour routing is applied in a fault condition in order to bypass faulty links and nodes, and it enables recovery of the network function (Figure 2b). PEACH continuously monitors the system and dynamically performs both adaptive routing as power and performance demand and detour routing to achieve a highly dependable network.

Though the number of nodes in PEARL is theoretically limitless, our design target is a sixteen-node network at the maximum. Figure 2c) shows an eight-node network example.

PEACH Architecture

Figure 3 shows the primary functional unit of PEACH. This chip includes two blocks, the control processing block and the transfer processing block, which are connected with a bus bridge. Figure 4 shows the chip micrograph. The features of PEACH are described in Table 1.

The transfer processing block has four PCIe ports each of which can transfer packets using up to four lanes, 512KB SRAM which is allocated for temporary packet storage, and an Intelligent ICU [5] in close liaison with the multicore processor and the PCIe I/F. All the main modules in the transfer processing block are connected by a high-speed internal system bus. The Intelligent ICU also supports the fast automatic data transfer function that offloads interrupt services from the multicore processors.

The control processing block performs data processing and data flow control, which consists of adaptive network routing and packet header analysis. In the control processing block, the eight cores are connected to a common pipelined bus with a cache coherence mechanism [6]. Each core is synthesizable and includes a FPU, a MMU, three 8KB memories for L1 Caches (I/D), and a local memory. The control processor is a symmetric multiprocessor (SMP) and supports a core grouping mode that divides cores into several groups [7]. The pipelined bus is connected to a 512KB L2 cache. This pipelined bus with a large bus width (a 128-bit read bus and a 32-bit write bus, separately) reduces its bus traffic and is directly connected to an internal multi-layer bus. The 256MB DDR3-600 I/F is accessed via both the control processing block and the transfer processing block in parallel. This high-speed and large memory contributes to improving the chip performance. The DDR memory is also used as a large packet buffer if

the packet size is larger than 512KB SRAM.

The multicore processor in PEACH offers an effective paradigm for fast packet processing. In a multicore system, careful consideration needs to be given to the hardware and software architecture. Network packet processing from a specific PCIe port can be assigned to dedicated cores to bind specific tasks and specific cores. By effective distributed processing on a multicore processor, high traffic rates from multiple 20-Gb throughputs can be realized on multiple PCIe ports.

PEARL network route construction

A network manager runs as a userland program on each Linux on PEACH. Each PEACH has a routing table in a driver. PCIe notifies the daemon of a change of link status using sysfs interface on Linux.

As described later, there are two data transmission flows. Intelligent ICU mode can transfer data fast but it only has a fixed routing table. Processor mode provides a flexible routing for error handling or a startup sequence. On startup, a master node does a route search and makes the routing table under the assumption of Intelligent ICU mode. When a fault occurs, the multicore processor acts as a backup, or routing tables are overwritten to modify the route.

Power management of network system

Each daemon program on PEACH monitors network status and sends information to elected master node. The master node makes a power-aware order to network, and each network manager on PEACH changes PCIe link configuration. Based on application's demand, network link performance can be changed. The multicore processor in PEACH has roles to monitoring the network status by demon program, managing PCIe PHY performance using upconfiguration function (which will be seen in "PCIe upconfiguration function") and processor power-management such as clock-gating.

An important point here is that adopting a multicore processor can provide fine-grained control of PCIe configuration and reduce the power consumption of the overall system.

Multi-port multi-lane PCIe I/F with upconfiguration function

The features of the PCIe I/F are described in Table 1. Each PCIe port has a link controller, PHY, local DMA controller (DMAC), and local packet buffer RAM. The latest Rev.2.0 standard has transfer rate of up to 5.0Gbps, double that of the Rev.1.1 (2.5Gbps). Rev2.0 supports both 2.5Gbps/5.0Gbps transfer rates because of a compatibility with Rev1.x. Furthermore, a procedural step in going from 2.5Gbps up to 5.0Gbps is required. The total transfer rate to each destination is 20Gbps and the theoretical peak

bandwidth is actually 2GB/s due to 8bit/10bit encoding for the embedded clock and error detection. PEACH with four PCIe ports realizes a high-performance communication of 4x20Gbps and the power efficiency of 0.04W/Gbps.

InfiniBand DDR 4x has a high bandwidth of 20Gbps and low latency of 2μs. Subnet Manager provides automatic fault recovery [8]. However, overall system power consumption increases, because a controller chip and a switch each consume 3 to 5W/port. Multiple switches, which are necessary for fault tolerance, would run counter to reduction cost and low power consumption. The power efficiency of 4X InfiniBand is 0.083W/Gbps [9]. Thus, PEACH provides 51.5% better power efficiency than 4X InfiniBand (Table 2).

PCIe upconfiguration function

A restrictive power-aware control of link-by-link power cut-off is available in InfiniBand. In contrast, PCIe has an effective power-aware control that can change the number of lanes and the lane speed on-the-fly across the link and nodes. We use a PCIe upconfiguration function that allows the transfer rate and the number of lanes to be switched in response to a change in bandwidth by software (Figure 5). When the required transfer volume is higher, the PCIe port operates at the full of 20Gbps. When the transfer volume is lower, only one lane operates at 2.5Gbps for low power consumption.

The comparison of the PCIe PHY power consumption is shown in Table 3. In low power consumption mode, using the PCIe port of 2.5Gbps provides 76% less power consumption than that of 20Gbps. The maximum transfer rate using the PCIe port of 20Gbps provides 52% better power efficiency compared to that of the low power consumption of 2.5Gbps. Figure 6 shows power consumption of PCIe PHY at each requested transfer volume. When the required transfer volume is lower than 2.5 Gbps, using one PCIe port of 2.5Gbps provides the lowest power consumption. When the required transfer volume is larger than 2.5Gbps, using PCIe ports of 5.0 Gbps is worthwhile.

Intelligent Interrupt Controller

Figure 7 shows a block diagram of Intelligent ICU. The Intelligent ICU communicates with PCIe ports within the PEACH. It also communicates with adjacent PEACHs and nodes. Both communications use message passing via the high speed system bus and via PCIe. The Intelligent ICU can also send a Message Signaled Interrupt (MSI) packet to the adjacent node via PCIe. To notify PCIe-Link/DMA transfer completion and PCIe errors, interrupt requests are directly sent from PCIe-Link to Intelligent ICU.

The key features of the Intelligent ICU are the following.

a) Interrupt relay function

b) Inter-chip interrupt function

c) Fast automatic data transfer function

All functions are used in inter-node communication.

(a) The interrupt relay function relays interrupt requests from PCIe I/F in the transfer-processing block to the cores via ICU in the control-processing block in PEACH. These interrupt requests are sent as notifications of the completion of PCIe linkup or PCIe DMA transfer processing.

(b) An inter-chip interrupt function sends information such as notification of the completion of chip-to-chip data transfer. An adjacent chip connected to PEACH via PCIe can write a control register in Intelligent-ICU to assert an interrupt request to a core.

(c) Fast automatic data transfer function automatically handles transfer processing without using cores in PEACH. Further discussion of this function is shown in Section “Smart Interrupt handling”.

Data Flow Control

IRQ affinity on Linux allows programs to specify which core services a given interrupt. In PEACH, IRQ affinity binds an interrupt from each PCIe port to a specific core in a one-to-one relationship. The network packet is directed to the desired core. By using this distributed processing, a packet processing can be done efficiently. Furthermore, a snooping group of cores alleviates snooping overhead, because cores can only be snooped from other cores in the same group. Eliminating unnecessary internal snoop transaction improves the stability of the communication services.

Figure 8a) illustrates the data flow control in PEACH. The solid dotted arrows indicate data flow. When PCIe #3 receives data from a node CPU, the data is temporarily stored in the SRAM. After that, the data is sent to another PEACH via the appropriate destination port (PCIe #0).

The red arrows indicate control flow. Devices connected to PEACH via PCIe can send control packets to the Intelligent ICU. The node CPU sends control packets and an interrupt request packet to establish communication. The Intelligent ICU relays this interrupt request to a core. In the interrupt handler, the core analyzes the packets, performs an address transformation, and launches the DMAC in PCIe.

Figure 8b) shows fault handling for dependability. When communication is broken up due to a fault on a link or an adjacent node, PCIe sends an interrupt request to a core via the Intelligent ICU. The core starts error recovery of removing the defective lane or applying detour routing.

Smart interrupt handling such as quick error response and speedup of the fault handling are essential for dependability. Therefore, good load balancing and performance tuning requires control of where interrupt service is performed. IRQ affinity assigns a specific core to a PCIe port to process interrupt service tasks requested only by that PCIe port. The system software makes the core idle steadily except during an interrupt services. There is no overhead of a context switch from a previous process and the core can smoothly move to the interrupt processing, which speeds up the interrupt response time (Figure 8c).

Smart interrupt handling also supports the fast automatic data transfer function (Figure 8d). The Intelligent ICU can transfer data without using the cores' interrupt services by performing address transformation and handling the DMAC in PCIe automatically.

Figure 9 shows two data transmission flows, (a) processor mode using interrupt services (b) Intelligent ICU mode using fast automatic data transfer. This chart indicates data transmission flows from Node CPU0 to Node CPU1 via PEACH (A) and PEACH (B). All communication packets between nodes are sent via PCIe.

Figure 9a) shows the data transmission flow using core interrupt services. After Node CPU0 sends a data packet and a control packet, Node CPU0 sends an interrupt request packet to the Intelligent ICU in PEACH (A) to establish a communication channel. The Intelligent ICU relays the interrupt request and control packet to a core in PEACH (A). In the interrupt handler (core-A), the core analyzes the packets including source and destination addresses and size of data, and then update the packet headers and sends the packets to the destination.

The core launches the DMAC in PCIe and transfers data to PEACH (B). PCIe notifies the core of the completion of data transfer via the Intelligent ICU. In the interrupt handler (core-B), the core sends a control packet and an interrupt request packet to PEACH (B) and an end packet to Node CPU0. PEACH (B) acts in a similar manner to PEACH (A) and Node CPU1 finally receives data.

Though packet processing executed in the multicore processor is flexible, interrupt processing overhead cannot be avoidable. Intelligent-ICU has a routing table and makes automatic route computation, which can reduce transfer latency. Figure 9b) shows how the fast automatic data transfer function handles transfer processing without using the multicore processor. Node CPU0 sends an initiate data transfer request packet to PEACH (A). The Intelligent ICU in PEACH (A) launches the DMAC in PCIe and automatically transfers data to PEACH (B). The Intelligent ICU in PEACH (B) acts in a similar way, Node CPU1 finally receives data.

The fast automatic data transfer function of the Intelligent ICU can dramatically reduce transfer processing time by 20% under normal network operation (Figure 10).

Because of the adoption of a multicore processor and the Intelligent ICU, PEACH acts as an intelligent network device.

Evaluation System

Based on the component descriptions in previous sections, we have developed a two-node prototype of our PEARL network system (see Figure 1). Figure 11c) shows a photograph of a PCIe x4 host adapter board that has a PEACH, and PCIe external cable connectors [10]. The board also has a CompactFlash (CF) card slot, 4MB flash memory and two 128MB DDR3 memories. The CF card contains a Linux ext3 file system including Linux kernel 2.6.35 and is also used for storing log information. Linux runs on a multicore processor in PEACH on a stand-alone basis, booting by loading the Linux kernel image on the CF card. This host adapter board can be inserted into a PCIe slot on the mother board of a computing node (Figure 11a). The cpu hotplug on Linux can dynamically suspend and resume a core responding to system load, which are useful for power awareness.

The PCIe architecture consists of four discrete logical layers (Figure 11b). From the bottom up, they are the physical layer, the data link layer, the transaction layer, and the software layer.

The software layer generates read and write requests that are transported by the transaction layer. The transaction layer manages the transactions for communication, such as read/write to/from memory, message passing, or configuration. The data link layer is responsible for link management, including packet sequencing and data integrity, which includes error detection and error correction. The physical layer includes all circuitry, including a driver with impedance matching and input buffers, parallel-to-serial and serial-to-parallel conversion, and PLLs. Each PCIe port has a physical layer

controller (PHY), a data link layer controller (MAC), and a transaction layer controller as hardware modules.

Switching time of PCIe upconfiguration function

Measurement results of the PCIe upconfiguration function switching time are shown in Table 4. While the time required to increase the lane speed is 6.5 μs, the time required to decrease it is 3.8 μs, which is because PCIe PHY needs extra time to gain lane speed (Table 4a). While the time required to shrink the number of lanes is 4.6 μs, the time required to expand it that is almost 9.1 μs (Table 4b). The minimum latency of DMA transfer between PEACHs that is also measured in this evaluation system is 1.0 μs. Thus, power-aware control can be performed with fine-grained operation.

PEACH/PEARL evolves wide range of communications by extending packet transmission of PCIe to inter-node communication. The performance advantage, power awareness and high dependability of PEACH, are the result of the combination of PCIe, the Intelligent ICU and the multicore processor. We are currently improving and expanding firmware including drivers and user communication libraries. PEARL is a product of the research area of DEOS project [11]. The PEACH board is positioned as a hardware platform for DEOS project and is expected to be adopted in many dependable high-end embedded systems, which spurs upgrades to technology innovations in this area.

Acknowledgements

This work is supported by a JST/CREST program entitled “Computation Platform for Power-aware and Reliable Embedded Parallel Processing Systems”.

References

1. T. Hanawa, et al, “PEARL: Power-aware, Dependable, and High-Performance Communication Link using PCI Express”, 2010 IEEE/ACM International Conference on Green Computing and Communications, Dec. 2010
2. The InfiniBand architecture specification. InfiniBand Trade Association. http://www.infinibandta.org/specs/
3. PCI Express Base Specification, Rev. 2.0, PCI-SIG, Dec. 2006.

http://www.pcisig.com/
4. S. Otani et al., “An 80Gb/s Dependable Communication SoC with PCIExpress I/F and 8 CPUs”, ISSCC Dig., Feb.2011.
5. S. Otani et al., “An 80Gb/s Dependable multicore Communication SoC with PCI Express I/F and Intelligent Interrupt Controller” Cool Chips XIV , April.2011.
6. S. Kaneko et al., “A 600MHz single-chip multiprocessor with 4.8GB/s internal shared pipelined bus and 512kB internal memory,” IEEE Journal of Solid-State Circuits Vol.39, No.1, pp.184-193, Jan. 2004
7. H. Kondo et al., “Design and Implementation of a Configurable Heterogeneous Multicore SoC With 9 CPUs and 2 Matrix Processors”, IEEE Journal of Solid-State Circuits Vol.43, No.4, pp.892-901, Apr. 2008
8. OpenFabrics Enterprise Distribution (OFED). OpenFabrics Alliance. [Online], http://www.infinibandta.org/specs/
9. “QLogic TrueScale™ InfiniBand, the Real Value”, http://www.qlogic.com/
10. PCI Express External Cabling Specification, Rev. 1.0, PCI-SIG, Jan. 2007.
11. “Dependable Embedded OS R&D Center and DEOS Project” http://www.dependable-os.net/index-e.html

Sugako Otani is a processor architect of M32R and RX and a manager of technical development in the CPU Development Department at Renesas Electronics. Her research interests include networking, microprocessor architecture. Otani has an MS in Physics from Waseda University.

Hiroyuki Kondo is the director of the CPU Development Department at Renesas Electronics. He is the chief processor architect of M32R and RX. His research interests include microprocessor architecture and operating systems. Kondo has a BS in Physics from Kyoto University.

Itaru Nonomura is a staff engineer at Renesas Electronics, where he is an interconnect designer. His research interests include on-chip and off-chip interconnects. Nonomura has a BS in electrical engineering from Waseda University.

Toshihiro Hanawa is an associate professor at Faculty of Engineering, Information and Systems, University of Tsukuba. He is also a faculty fellow at Center for Computational Sciences, University

of Tsukuba. His research interests include high performance computing systems. Hanawa has a PhD in engineering from Keio University.

Shin'ichi Miura is a researcher at Center for Computational Sciences, University of Tsukuba. His research interests include system software and interconnects for high performance computing. Miura has a PhD in engineering from University of Tsukuba.

Taisuke Boku is a professor of Faculty of Engineering, Information and Systems, University of Tsukuba and is also the deputy director of Center for Computational Sciences, University of Tsukuba. His research interests include high performance architecture, large scale system software and low-power and high-performance communication. Boku has a PhD in engineering from Keio University.

Sugako Otani
System Core Development Div.
Renesas Electronics Corporation
4-1 Mizuhara Itami Hyogo, 664-0005, Japan
phone: +81-72-787-5244 fax: +81-72-789-3004
sugako.otani.uj@renesas.com

Hiroyuki Kondo
System Core Development Div.
Renesas Electronics Corporation
4-1 Mizuhara Itami Hyogo, 664-0005, Japan
phone: +81-72-787-5229 fax: +81-72-789-3004
hiroyuki.kondo.xm@renesas.com

Itaru Nonomura
System Core Development Div.
Renesas Electronics Corporation
5-20-1 Mizumotocho Kodaira Tokyo, 187-8588
itaru.nonomura.xb@renesas.com
phone: +81-42-312-6286 fax: n/a

Toshihiro Hanawa
Center for Computational Sciences
University of Tsukuba
1-1-1 Tennoudai Tsukuba, Ibaraki, 305-8577
hanawa@ccs.tsukuba.ac.jp
phone: +81-29-853-6290 fax: n/a

Shin'ichi Miura
Center for Computational Sciences
University of Tsukuba
1-1-1 Tennoudai Tsukuba, Ibaraki, 305-8577
miura@hpcs.cs.tsukuba.ac.jp

phone: +81-29-853-6487 fax: n/a

Taisuke Boku
Center for Computational Sciences
University of Tsukuba
1-1-1 Tennoudai Tsukuba, Ibaraki, 305-8577
taisuke@cs.tsukuba.ac.jp
phone: +81-29-853-6487 fax: n/a

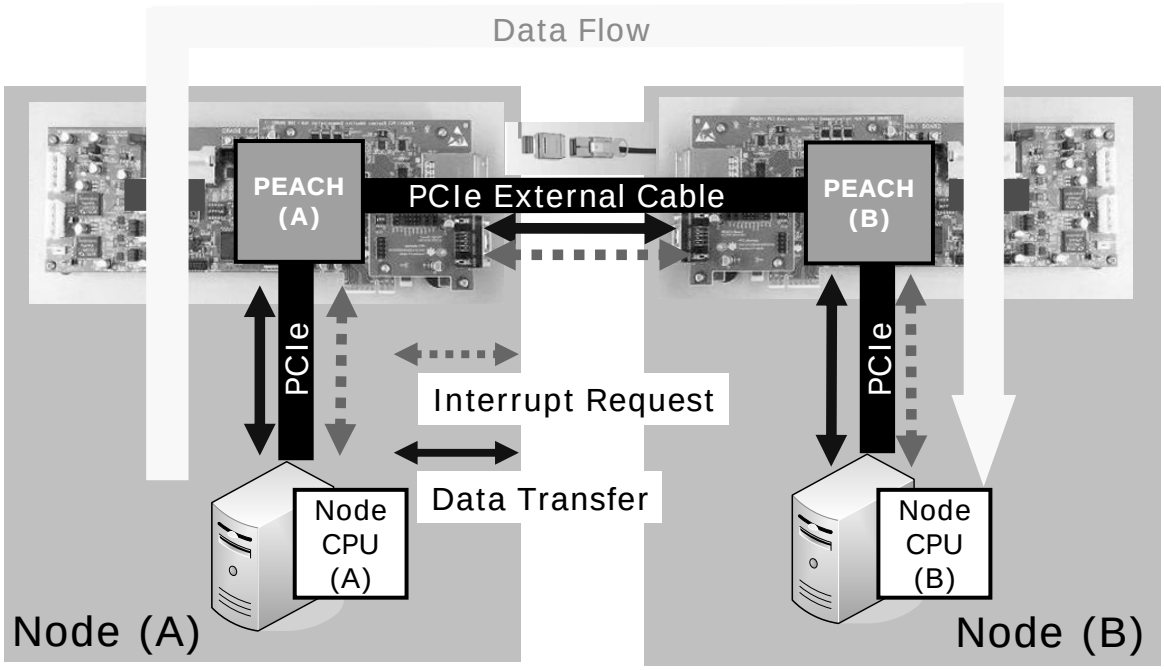


Figure 1. The communication link, PEARL, connects computing nodes with a PCIe external cable. A network interface card with the network device, PEACH, can be inserted into a PCIe slot on a mother board of a computing node.

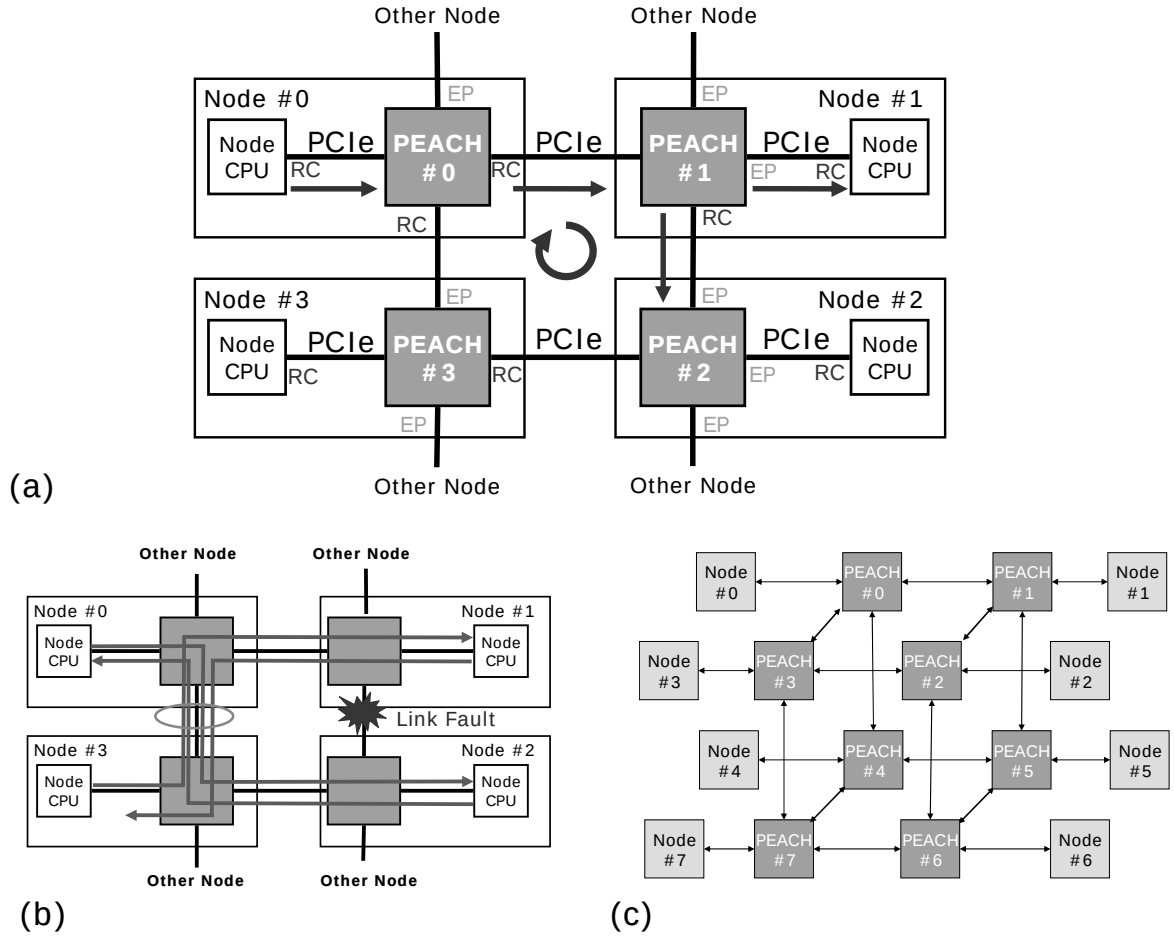


Figure 2. (a) shows neighbor communication on PEARL. (a) and (b) show adaptive routing under a normal network condition and detour routing in a fault network condition. (c) shows an example of an eight-node network.

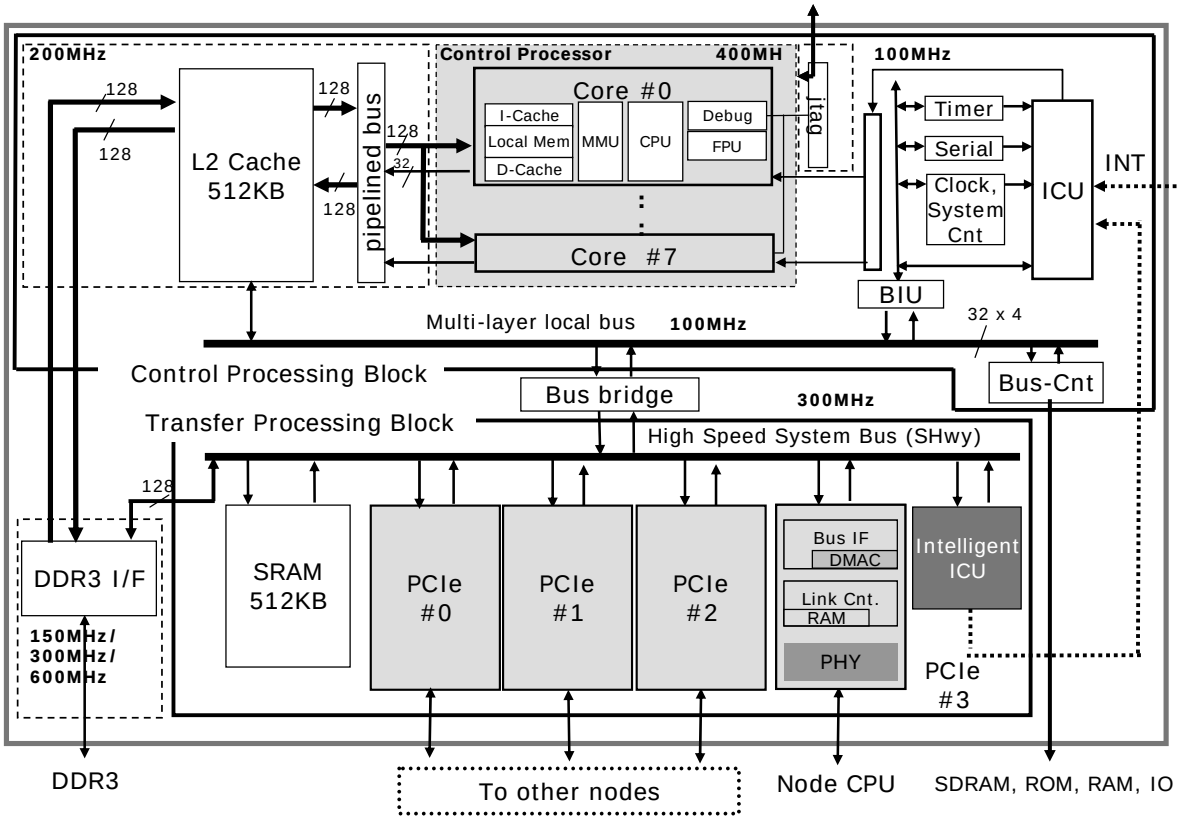


Figure 3. PEACH block diagram. This chip integrates an eight-core control processor, four PCIe ports, and an intelligent ICU.

Chip Characteristic	Description
Clock frequency	Internal: 400 MHz max. External bus: 100 MHz
Processor	8core, SMP L1-cache:8kB(I)+8kB(D), LM:8kB, MMU, FPU
Core	32-bit Processor (400 MHz max.)
Memory	L2 cache: 512 kB Internal SRAM: 32 kB, 512 kB
DRAM I/F	DDR3-600 I/F x 1, SDRAM I/F x 1
PCIe I/F	PCI Express standard Rev.2.0 Transfer speed: 5.0 GT/s, 2.5 GT/s per lane 4 lanes (20 Gbps) x 4 ports Upconfiguration function Automatic retransmission function Selectable Root Complex / Endpoint
Intelligent Interrupt Control Unit	Transfer address, size information register x 3 Initiate data transfer function
Bus	Packet router Multi-layer bus (4-layer) Pipelined bus

Table 1. Chip Features

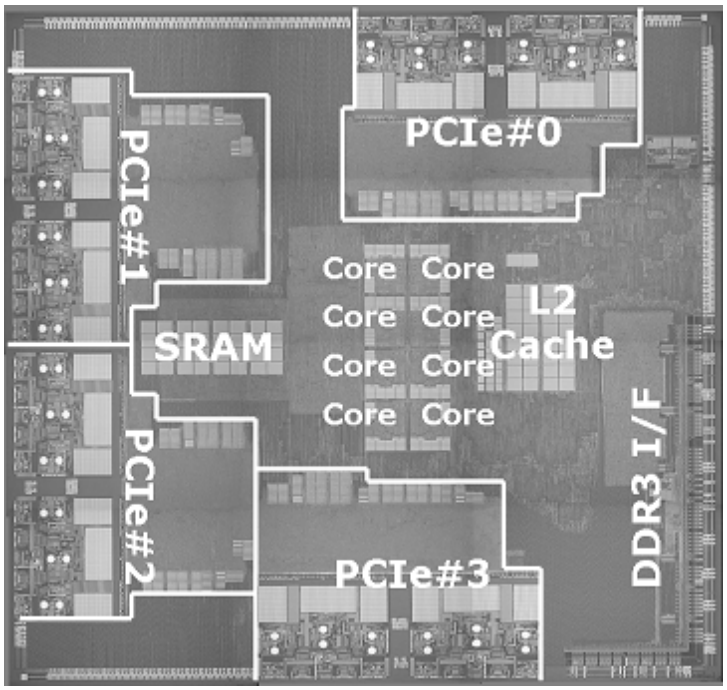


Figure 4. PEACH micrograph. The test chip was fabricated in a 45nm low-power CMOS (8 layers, triple-Vth).

	4x InfiniBand	PEARL
Network Device	Dedicated Circuit InfiniBand	PEACH PCI Express
Power Efficiency [W/Gbps]	0.083	0.040

Table 2. Comparison of Power Efficiency

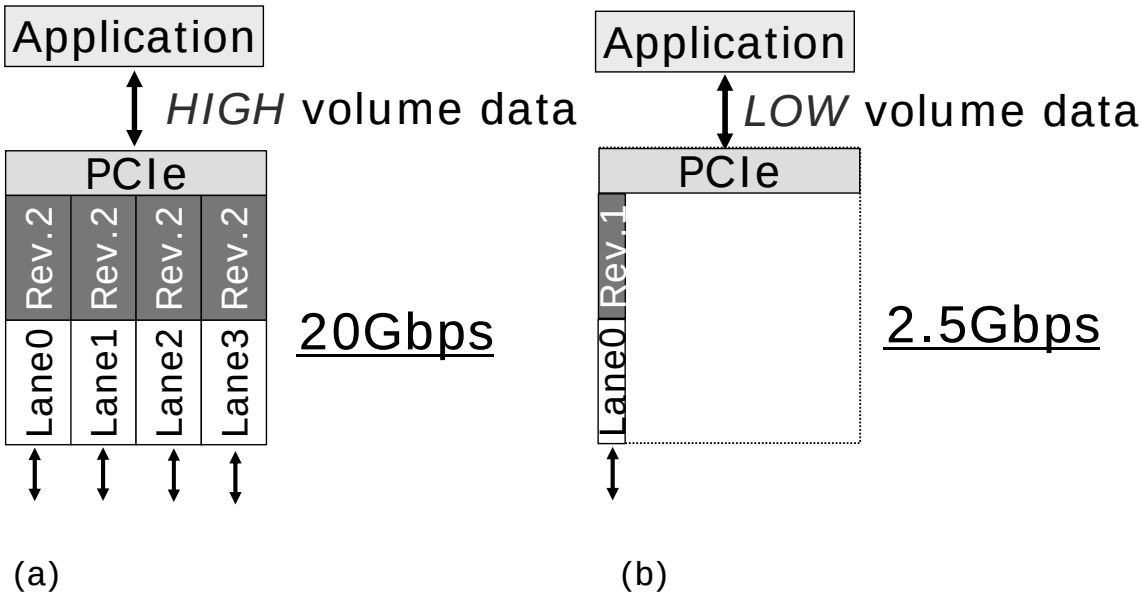


Figure 5. PCIe upconfiguration function by software control. (a) Maximum data transfer rate (b) Low power consumption

	No. of lanes		
Lane Speed	**4 lanes**	**2 lanes**	**1 lane**
5Gbps	**20Gbps** 1.00	0.50	0.28
2.5Gbps	0.84	0.42	**2.5Gbps** 0.24

Power Efficiency of PCIe PHY (W/Gbps)

(1.00/20) / (0.24/2.5) = ***0.52***

5Gbps@4 lanes 2.5Gbps@1 lane

Table 3. Power Consumption of PCIe PHY (Normalized)

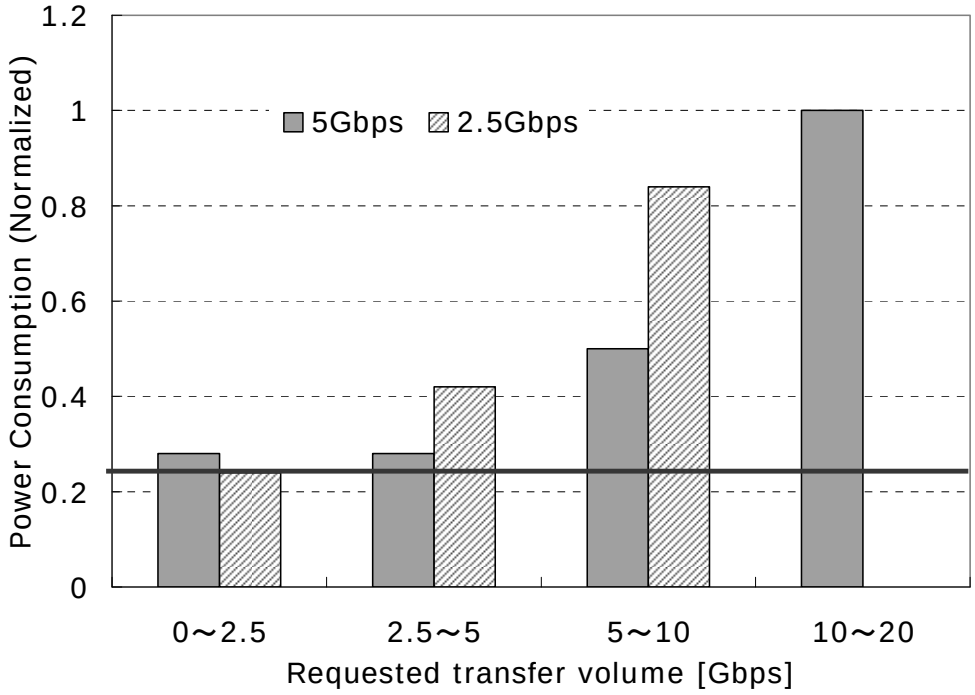


Figure 6. Power Consumption of PCIe PHY (W) at each requested transfer volume.

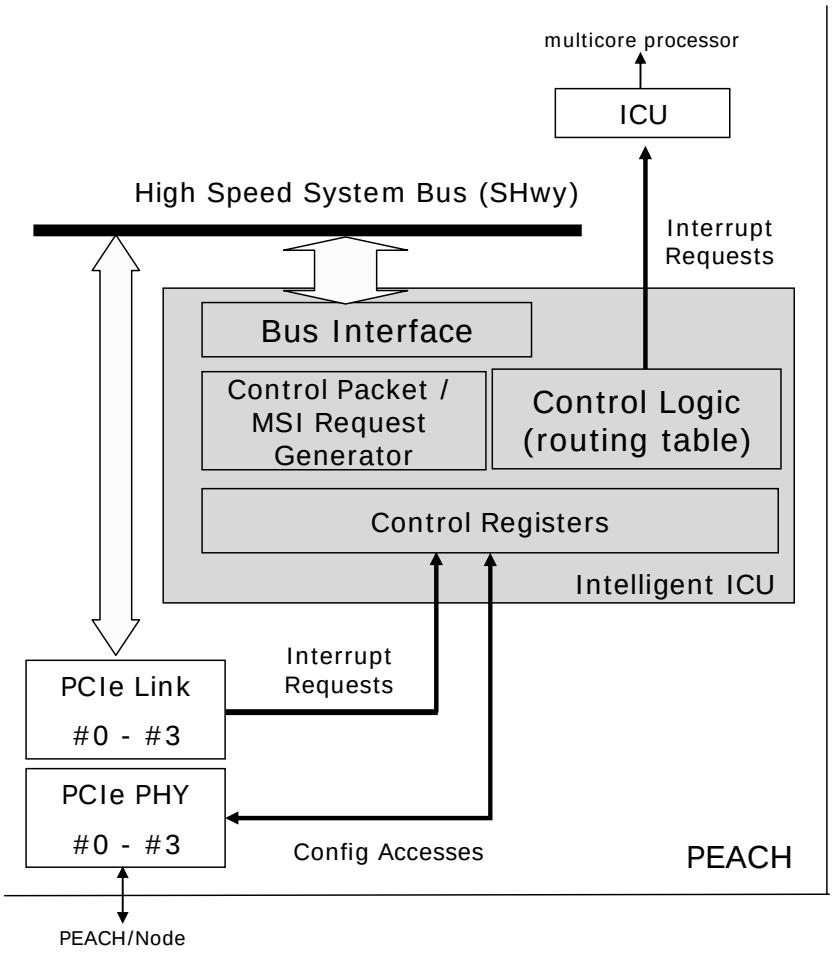


Figure. 7. Block Diagram of Intelligent ICU

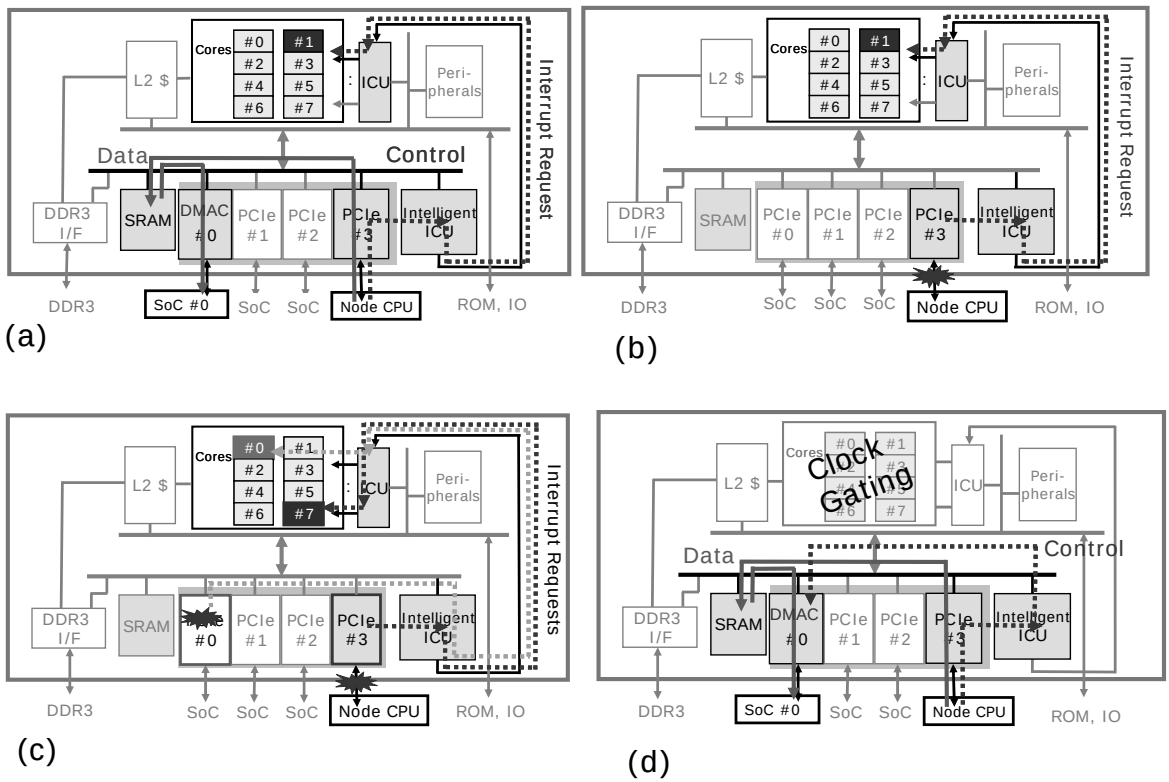


Figure 8. (a) shows data flow control in PEACH. (b) shows fault handling. Intelligent ICU relays a change of PCIe link status to the cores. (c) and (d) shows smart interrupt handling: (c) illustrates IRQ affinity which binds an IRQ from each PCIe port to a specific core. (d) illustrates fast automatic data transfer function that offloads interrupt services from the cores.

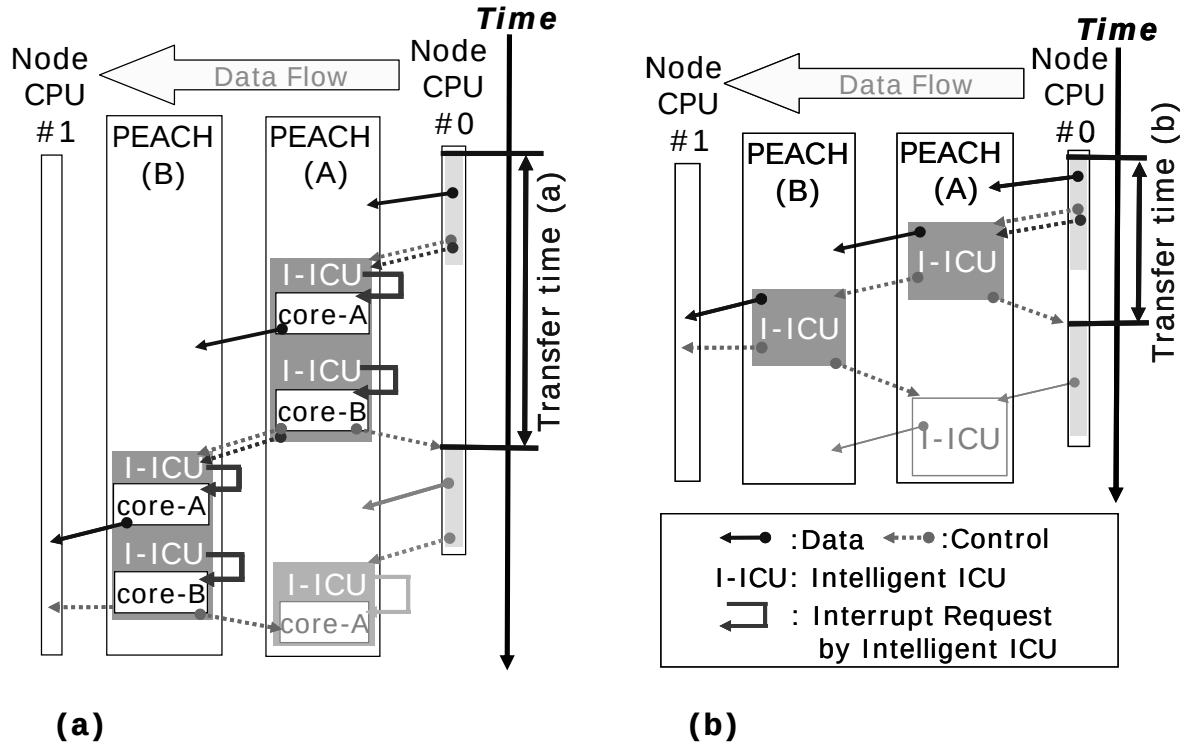


Figure 9. Two data transmission flows: (a) Processor mode using interrupt services (b) Intelligent ICU mode using fast automatic data transfer.

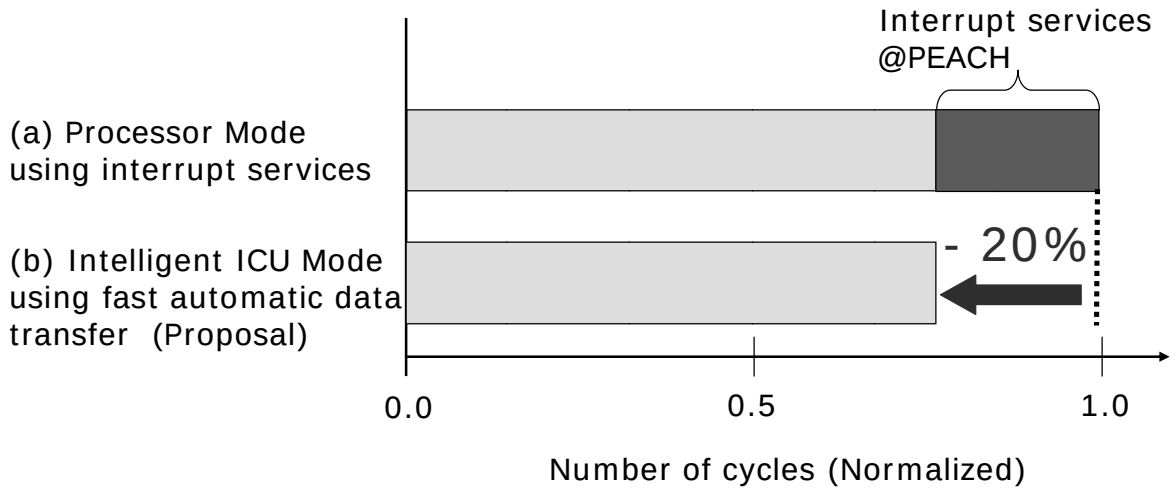
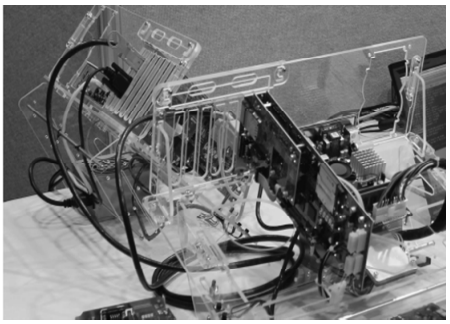
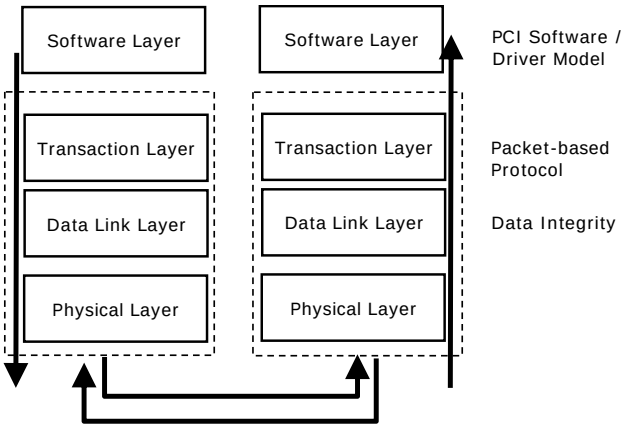


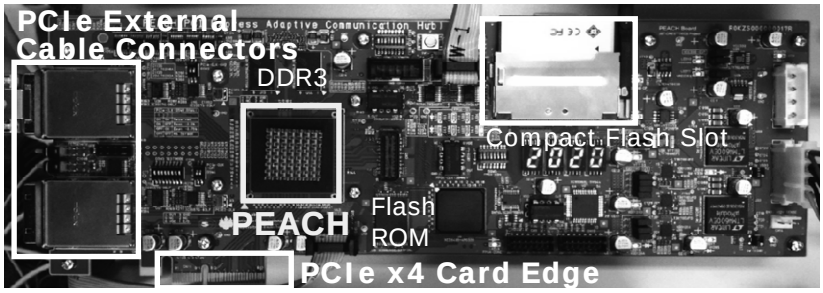
Figure 10. Fast automatic data transfer function of the Intelligent ICU improves transfer latency.



(a)



(b)



(c)

Figure 11 shows the PEARL evaluation system (a), and a PCIe x4 host adapter board (c). (b) illustrates PCIe logical layers.

(a) Switching time of the lane Speed

Lane Speed	Time [us]
2.5Gbps → 5.0Gbps	6.5
5.0Gbps → 2.5Gbps	3.8

(b) Switching time of the number of lanes

No. of lanes	Time [us]		
To \ From	1	2	4
1	---	4.6	4.6
2	9.1	---	4.6
4	9.1	9.0	---

Table 4 upconfiguration function switching time of PCIe